

UNIVERSIDADE FEDERAL DO PARANÁ

DIEGO LINKE

FRAMEWORK DE GESTÃO DE PROJETOS COMPLEXOS

PARA IMPLEMENTAÇÃO DE SOFTWARE

CURITIBA

2012

DIEGO LINKE

FRAMEWORK DE GESTÃO DE PROJETOS COMPLEXOS
PARA IMPLEMENTAÇÃO DE SOFTWARE

Monografia apresentada ao curso de MBA de Gestão Estratégica como requisito parcial de conclusão do curso da Universidade Federal do Paraná.

Orientador: Prof. Dr. Cleverson
Renan da Cunha

CURITIBA

2012

LISTA DE ILUSTRAÇÕES

FIGURA 1: RESTRIÇÃO TRIPLA DE GERENCIAMENTO DE PROJETO (PMI, 2004, p. 375).....	9
FIGURA 2: RESTRIÇÃO TRIPLA DEFENDIDA POR KERZNER (2011)	10
FIGURA 3: CLASSIFICAÇÃO DE PROJETOS (STACEY, 2000).....	16
FIGURA 4: FRAMEWORK SCRUM (WOODWARD, 2010).....	20
FIGURA 5: VISÃO GERAL DO FRAMEWORK.....	22
FIGURA 6: CERIMONIAS DO FRAMEWORK	26
FIGURA 7: EXEMPLO DE GRÁFICO DE ACOMPANHAMENTO – BURNDOWN.....	29
FIGURA 8: EXEMPLO DE TASK BOARD (SIMS; JOHNSON, 2011)	30

LISTA DE SIGLAS

OGC® - *The Office of Government Commerce, UK.*

PMBOK® – *Project Management Body of Knowledge.*

PMI® – *Project Management Institute, Inc.*

PRINCE2™ - *Projects In Controlled Environments*

TSO – *The Stationery Office* – www.tso.co.uk

LISTA DE ABREVIATURAS

Dr. – Doutor

Prof. – Professor

SUMÁRIO

1. INTRODUÇÃO	7
1.1. Contexto	7
1.2. Identificação do Problema	7
1.3. Objetivo da Pesquisa	11
1.4. Justificativas	12
2. Referencial teórico	13
2.1. O Scrum	18
3. Framework de gerenciamento de projetos complexos.....	21
3.1. Papéis	23
3.2. Cerimonias	24
3.3. Artefatos	27
4. Alinhamento Comercial	33
5. Conclusão	35

1. INTRODUÇÃO

1.1. Contexto

O mercado de tecnologia é muito dinâmico, caracterizado por mudanças constantes e atualizações tecnológicas incessantes, que por sua vez resultam em diversos problemas, mas também em muitas oportunidades.

Realizar projetos de implementação de softwares neste cenário torna-se um grande desafio, principalmente com a área de tecnologia se aproximando mais da área de negócio das empresas, transcendendo o seu escopo de suporte ao negócio para uma abrangência mais estratégica tornando-se diferencial competitivo.

Este trabalho propõe um framework de gerenciamento para projetos complexos de implementação de softwares, considerando e criando meios para responder a mudanças constantes, falta de definição de escopo, entrega de valor antecipada e incremento de satisfação do cliente.

O foco será implementação de softwares de gerenciamento de TI, porém o framework poderá ser aplicado em outras categorias de softwares, tais como softwares de gestão de negócio, automação, entre outros.

1.2. Identificação do Problema

Usualmente empresas que desenvolve soluções de softwares não específicos para um cliente, possuem duas áreas de projeto, a primeira voltada para o desenvolvimento do software em si e a segunda com foco na implementação do software, o que muitas vezes é chamada de *service delivery*, consultoria ou serviços profissionais.

Esta segunda área também é encontrada em empresas de consultoria e integradoras. Este trabalho tem foco única e exclusivamente na implementação de software e não no seu desenvolvimento.

Tipicamente o processo de um projeto de implementação de software inicia-se na área comercial, quando o gerente de contas faz as apresentações das vantagens, benefícios e características do software. Durante este processo normalmente é envolvido um consultor de pré-vendas que detém o conhecimento técnico e comercial o suficiente para avaliar a efetividade da aplicação da solução no cliente alvo, ajudá-lo a justificar o investimento, identificar necessidades corporativas, definir a abrangência, arquitetura, módulos necessários, customizações e criar um esboço do escopo do projeto.

Ainda durante o processo de venda, a equipe de serviços profissionais é acionada para detalhar melhor o escopo do projeto, realizar um levantamento no cliente, estimar as customizações, entender as prioridades, definir os padrões de qualidade, identificar riscos, criar um cronograma e um plano de trabalho detalhado.

Após a finalização deste trabalho a área comercial poderá entregar ao cliente uma proposta contendo o preço do software e o custo de implementação da solução.

Seria perfeito se todos os projetos de implementação de softwares seguissem este processo e tivesse um detalhamento do escopo, levantamento dos riscos, cronograma detalhado, padrões de qualidade e fosse possível estabelecer um plano de projeto como descrito nas melhores práticas de gestão de projetos.

Infelizmente a prática nos mostra que nem sempre é assim, sendo o contrato assinado sem detalhamento do ambiente, com o escopo mal definido e com expectativas divergentes. Alguns motivos para que isto ocorra são:

- Pressão corporativa para fechar o negócio o quanto antes;
- Pressão do cliente para fechar o negócio para poder usar orçamento disponível remanescente;

- Falta de agenda dos consultores responsáveis para realizar este trabalho, versus restrições de tempo do projeto acabam pressionando o seu fechamento precipitado;
- Cliente não sabe exatamente o que gostaria ou precisa;
- Ausência ou pouco suporte corporativo gerencial do cliente;
- Cliente não tem detalhamento do ambiente;
- Expectativa e objetivos finais do cliente não são devidamente esclarecidos;
- Cliente não detalha todo o seu ambiente, objetivando minimizar o projeto e subestimar os custos de implementação;
- Desavenças internas entre áreas no cliente (ex: Arquitetura, Desenvolvimento e Produção) geram lacunas de informação;
- Equipe do cliente designada para trabalhar no projeto não compreende o valor para o negócio;

Além dos cenários expostos acima, muitos projetos sofrem interferência posteriores e a consequência disto é muitas mudanças, alteração de prioridades, incertezas dos objetivos, falta de alinhamento com empresas terceiras envolvidas no projeto, entre outros problemas.

Como resultado, ao menos uma das três variáveis da restrição tripla de gerenciamento de projeto, tempo, escopo e custo é afetada diretamente.

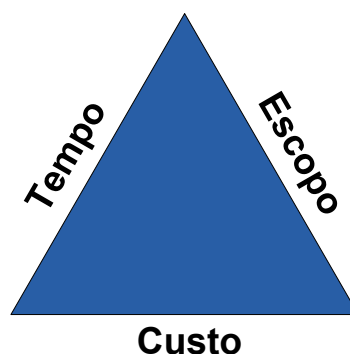


Figura 1 Restrição tripla de Gerenciamento de Projeto (PMI, 2004, p. 375)

Estes cenários expostos acima tornam um projeto complexo. HASS (2009) defende que “Projetos de TI são quase sempre complexos e difíceis de gerenciar. Uma fonte comum de complexidade em projetos de TI normalmente inicia-se com informações limitadas, quando os benefícios do negócio para serem alcançados e o que é necessário para alcançá-los não são conhecidos. Outro problema surge quando o escopo é adicionado durante o processo de desenvolvimento simplesmente porque não foi possível definir todos os requerimentos inicialmente.”

Além disto, outro fator importantíssimo é a satisfação do cliente. Muitas vezes o projeto é entregue no prazo, com o escopo e qualidade especificada, com o custo acordado, mas o cliente fica insatisfeito porque ele não conseguiu gerar o valor esperado para o negócio, causando frustração e descontentamento.

Alguns autores como KERZNER (2011) defendem criar um círculo em volta da restrição tripla de projetos incluindo a satisfação do cliente, conforme ilustração a seguir.



Figura 2: Restrição tripla defendida por KERZNER (2011)

Isto tipo de abordagem é importante e se faz necessário porque as empresas continuam focando em fechar o escopo dos projetos e limitar-se em entregar o que foi acordado, independente se o que o cliente deseja é mais simples ou não.

1.3. Objetivo da Pesquisa

O objetivo deste trabalho é propor um framework de gerenciamento para projetos complexos de implementação de softwares para lidar com estes cenários expostos.

Para alcançar este objetivo será adaptado um conhecido framework de desenvolvimento de software ágil chamado SCRUM (SCHWABER; SUTHERLAND, 2012) em um modelo interativo, adaptável, baseado em trabalhos incrementais e alinhado as expectativas do cliente para projetos de implementação de software, buscando:

- Flexibilidade
- Governança
- Antecipação de entrega de valor
- Interatividade
- Alinhamento com objetivos corporativos
- Incremento de satisfação dos clientes

As etapas para que este objetivo seja alcançado serão:

- Entender por completo o framework SCRUM seus processos, eventos e artefatos;
- Adaptar a terminologia voltada para desenvolvimento de software para um contexto de projeto de implementação de software;
- Excluir ou adaptar o que não for aplicável a projetos de implementação de software;
- Garantir que o processo de lição aprendida, através de interações múltiplas e revisões constantes sejam mantidos, assegurando assim um dos pilares do framework SCRUM que é a adaptação e melhoramento continuado.
- Incluir artefatos provenientes de outras áreas de conhecimento de gestão de projetos para compor um framework conciso e alinhado aos objetivos supracitados;

- Assegurar-se que o framework resultante seja adaptável e flexível a novos processos e artefatos que se fizerem necessário.

1.4. Justificativas

A justificativa para que este trabalho seja desenvolvido é a falta de padronização, ausência de cobertura específica do tema e principalmente necessidade de mercado baseada em experiências do autor. Dentre os motivadores destacam-se:

- Desmitificar e criar um framework capaz de gerenciar projetos sem escopo bem definido de forma eficiente e com alto índice de satisfação do cliente;
- Lidar com as constantes mudanças de cenários, ambientes e objetivos;
- Estreitar o relacionamento com o cliente, visando maior envolvimento e comprometimento e por consequência incremento do índice de sucesso dos projetos;
- Criar alternativa de metodologia gestão de projetos em situações adversas, com alto grau de exposição a riscos, excessivo dinamismo e falta de definições específicas que tradicionalmente criam lacunas de conhecimento e planejamento falhos;
- Entregar o benefício ao cliente antecipadamente e com menor custo possível;
- Gerenciar processos empíricos;
- Diminuir o impacto negativo de um projeto mal definido e aumentar a sua probabilidade de sucesso e satisfação do cliente;
- Evitar projetos intermináveis com baixa rentabilidade, ou até mesmo prejuízos, causados por falta de definição de escopo e mudanças constantes não gerenciadas;
- Estabelecer processos cíclicos e aplicáveis de melhoramento continuado capaz de gerar valor e resultado perceptíveis a empresa e ao cliente.

2. REFERENCIAL TEÓRICO

Atualmente o termo projeto é cada vez mais citado nas organizações, algumas vezes de maneira incorreta referenciando-se a uma rotina operacional já estabelecida.

Visando esclarecer este assunto, vamos inicialmente analisar o que é um projeto do ponto de vista dos principais referencias do mercado.

Para o PMI® um “projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A sua natureza temporária indica um início e um término definidos. O término é alcançado quando os objetivos tiverem sido atingidos ou quando se concluir que esses objetivos não serão ou não poderão ser atingidos e o projeto for encerrado, ou quando o mesmo não for mais necessário. Temporário não significa necessariamente de curta duração. Além disso, geralmente o termo temporário não se aplica ao produto, serviço ou resultado criado pelo projeto; a maioria dos projetos é realizada para criar um resultado duradouro” (PMI, 2008, p. 5).

Outra boa definição sobre projeto vem da metodologia PRINCE2™ “Um projeto é uma organização temporária que é criada com o propósito de entregar um produto de acordo com o *Business Case* acordado” (OGC, 2009, p. 1).

Note que ambas as definições acabam ressaltando a palavra “temporário” o que de certa forma é a principal diferenciação de uma rotina operacional. Como exemplo podemos citar a implantação de uma linha de fábrica. A implantação da linha de fábrica é um projeto tendo um inicio, objetivo, execução, entrega e fim, a operação de produção dia-a-dia após ela ter sido implementada é uma rotina operacional.

Uma vez definido o que é um projeto, temos que entender exatamente o que é um framework de gestão de projetos. O framework de gestão de projetos é um conjunto de processos e técnicas preestabelecidos que visa garantir a eficiência do trabalho realizado e entrega do produto do projeto dentro dos padrões de qualidade, custos e

prazo definidos. O termo framework foi escolhido, ao invés de metodologia, pois se trata de um conjunto de processos, ferramentas e melhores práticas onde a customização e adaptação são aceitas, recomendada e incentivada a fim de refletir a realidade de outras empresas bem como atender as demandas e necessidades de mercado específicas.

Projetos de softwares são naturalmente complexos para serem especificados, desenvolvidos e serem alinhados integralmente com os objetivos de negócios, expectativas e anseios dos seus patrocinadores.

Em seu livro *“Software Project Secrets. Why Software Projects Fail”* STEPANEK (2005, p. 8) lista fatores que justificam o porque projetos de softwares se diferem de outros projetos tornando-o muito mais arriscado, difícil de gerenciar e mutáveis:

- Software é complexo;
- Software é abstrato;
- Requerimentos são incompletos;
- Tecnologia muda rapidamente;
- Melhores práticas não são maduras (ou em constante evolução);
- Tecnologia é um assunto vasto;
- Experiência em tecnologia é incompleta;
- Desenvolvimento de software é uma pesquisa;
- Trabalho repetitivo é automatizado;
- A construção é na verdade o design;
- Mudanças são consideradas fáceis;
- As mudanças são inevitáveis.

O modelo tradicional de desenvolvimento de software é também conhecido como *Waterfall*, onde o desenvolvimento e a entrega do projeto de software são divididos em processos macros como (SIMS; JOHNSON, 2011):

1. Coleta de requerimentos

2. Design (Planejamento)
3. Desenvolvimento
4. Teste

Cada passo deverá estar completo antes de mover para o próximo e todos os processos deverão estar finalizados antes de qualquer entrega, geração de valor, ser feita a organização.

Este modelo de desenvolvimento de software é totalmente guiado, e tem seu sucesso atribuído, diretamente a qualidade dos requerimentos identificados. A qualidade destes estão vinculados a atributos que tipicamente constituem um bom requerimento (PRIES; QUIGLEY 2011):

- Rastreável
- Coesivo
- Completo
- Consistente
- Correto
- Atual
- Externamente observável
- Inequívoco
- Mandatório
- Verificável

Fazer um bom levantamento do cenário, riscos, requerimentos, restrições e escopo do projeto é sempre um desafio principalmente quando existe interesses divergentes, urgência, pressões entre outras situações já relatadas neste trabalho.

Quando correlacionamos o grau de acordo entre as partes e detalhe do escopo do projeto com o conhecimento da tecnologia que será utilizada, podemos classificar projetos como simples, complicado e complexo.

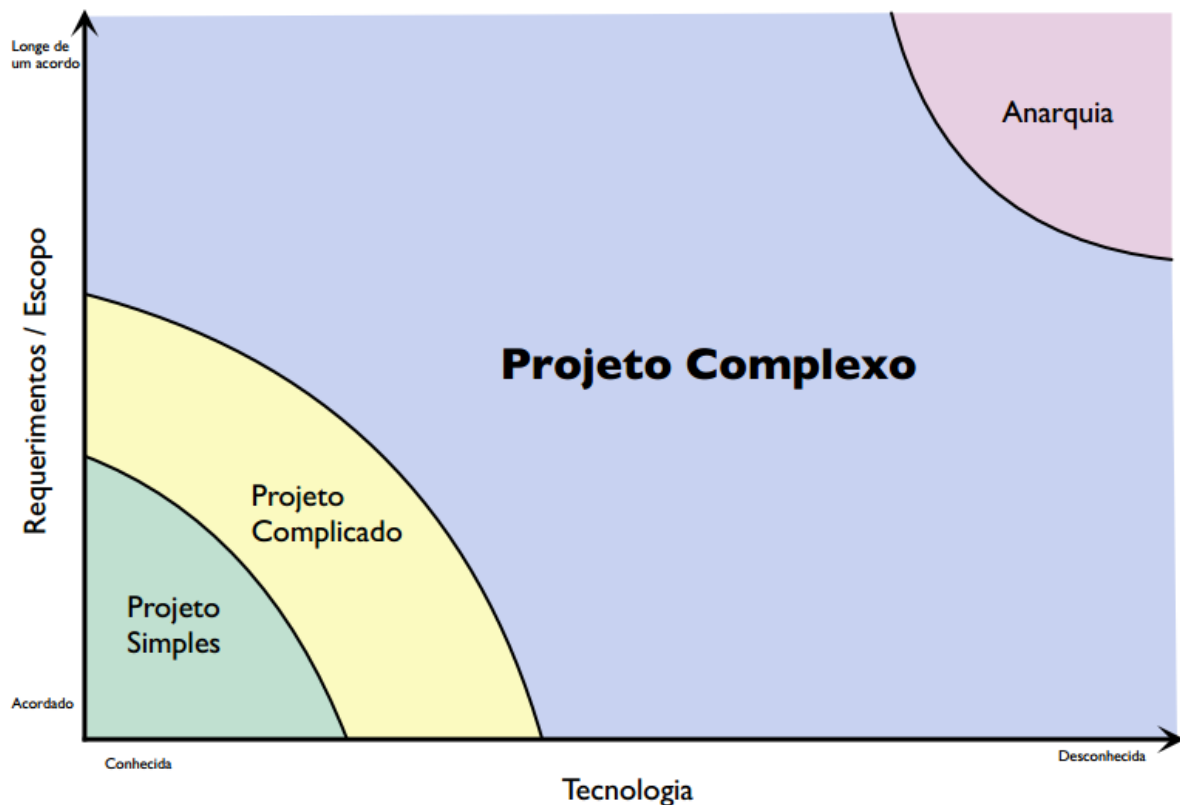


Figura 3: Classificação de projetos (STACEY, 2000)

Principalmente para trabalhar neste grande espaço relacionado a projetos complexos que Ken Schwaber e Jeff Sutherland criaram o Scrum e testaram e aprimoraram em suas organizações, dentre elas Individual, Inc., Fidelity Investments e IDX (atual GE Medical). O Scrum foi apresentado ao público pela primeira vez em um conferencia OOPSLA em 1995 (SCHWABER; SUTHERLAND, 2012, p. 17).

Porém não foi apenas Ken Schwaber e Jeff Sutherland que começaram a pensar diferente, este processo de metodologia mais empírica, incremental, voltada a entrega de valor o mais breve possível. Estas características foram instigando outras pessoas da área de software e novos frameworks foram sendo criados.

No início de 2001 durante 2 dias 17 pessoas, dentre elas Ken Schwaber e Jeff Sutherland, se reuniram em Snowbird Ski Resort em Wasatch Mountains – Utah para tentar chegar em um acordo sobre as melhores práticas de desenvolvimento de software. Porém estas pessoas não chegaram em um acordo sobre todos os

tópicos, mas concordaram em 4 importantes pontos, que vivaram o *Agile Software Development Agile Manifest*, que foi assinado por todos os participantes (HIGHSMITH, 2001).

Estes pontos norteiam todas as metodologias ágeis de desenvolvimento e estão publicados no sítio agilemanifesto.org em 44 línguas, dentre elas o Português do Brasil. Manifesto para Desenvolvimento Ágil de Software:

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.” (SCHWABER; SUTHERLAND, 2001).

Desenvolvimento Ágil bem como o framework Scrum cresceram exponencialmente desde então. Estudos recentes concluíram que times que utilizaram metodologias ágeis conseguem ser em média 37% mais rápido para entregar o software para o mercado e 16% mais produtivo, quando comparado a média da indústria (QSM, 2008).

Com estes resultados expressivos, a flexibilidade, capacidade de gerenciar mudanças e mitigar riscos pelo incremento de interações foram os grandes motivadores para que este trabalho seja desenvolvimento e aplicar este framework

adaptado para outros contextos como, por exemplo, o gerenciar projetos de implementação de software.

Tony Riches, também foi além do escopo de desenvolvimento de software e definiu o termo Gerenciamento de Projetos Ágeis como “uma abordagem prática que pode nos ajudar a gerenciar mudanças consistentemente e efetivamente. Isto incluem todos as ideias de senso comum dos métodos de gerenciamento de projetos desenvolvidos a muitos anos, mas reconhece que nós temos que ser flexíveis, adaptáveis e ágeis” (RICHES, 2012).

2.1. O Scrum

O termo Scrum varia do nome de uma jogada de Rugby onde duas equipes disputam a posse da bola. O Scrum é um framework criado originalmente para desenvolvimento de software e está cada vez mais sendo utilizado para desenvolver e manter projetos de produtos complexos.

O Scrum é um framework bem simples, mas muitas vezes complexos de ser implementado. Basicamente é composto por 3 papeis, 4 cerimoniais e 3 artefatos (SCRUM ALLIANCE, 2012).

Papéis:

- **Product Owner:** Responsável pelo valor ao negócio do projeto .
- **ScrumMaster:** Responsável por garantir que o time é funcional e produtivo.
- **Team:** Time auto-organizável para realização dos trabalhos

Cerimonias:

- **Sprint Planning:** Reunião do time com o Product Owner para definir o que será feito dentro da uma *Sprint*.
- **Daily Scrum:** Reunião diária do time para discutir impedimentos e evolução.

- **Sprint Review:** Time faz a demonstração para o Product Owner o que foi feito durante a Sprint.
- **Sprint Retrospective:** O time faz uma reavaliação e propõem mudanças para melhorar o produto e o processo de trabalho.

Artefatos:

- **Product Backlog:** Lista priorizada de funcionalidades desejadas do produto, mantidas pelo Product Owner.
- **Sprint Backlog:** Conjunto de trabalho proveniente do Product Backlog acordado pelo time para ser desenvolvido na Sprint detalhado por atividades.
- **Gráfico Burndown:** Gráfico de acompanhamento do andamento da Sprint baseado no tempo de trabalho restante.

A ilustração a seguir demonstra a distribuição destes principais componentes dentro da estrutura do framework Scrum.

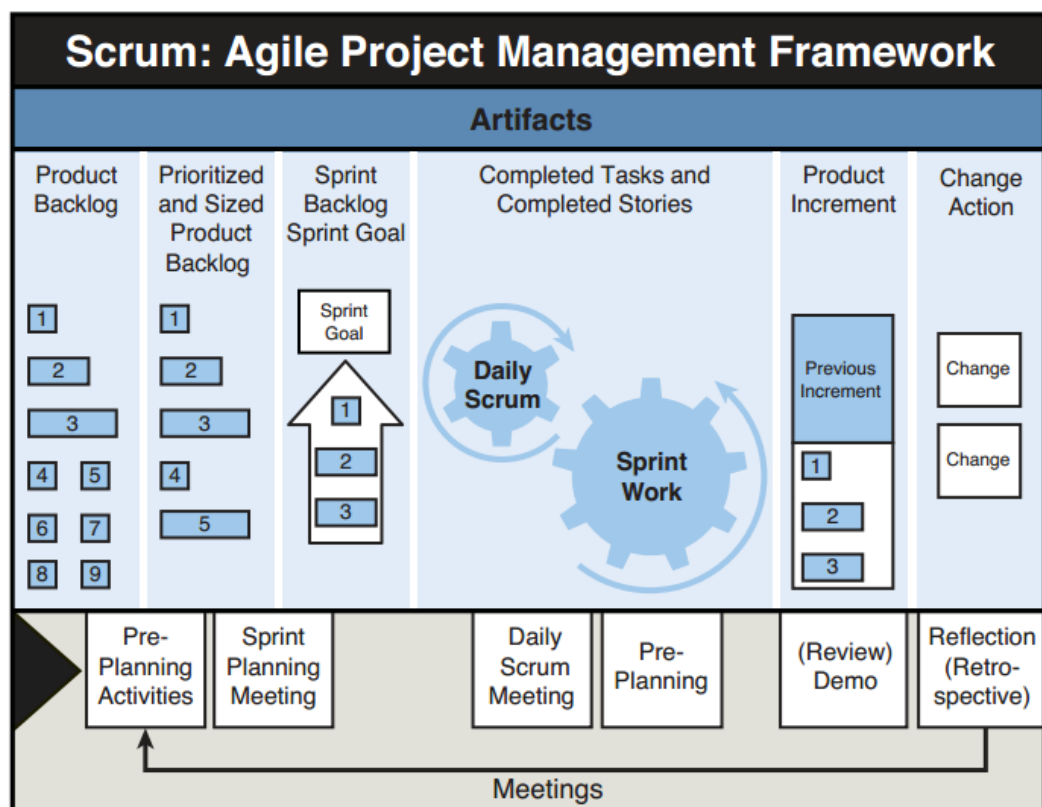


Figura 4: Framework Scrum (WOODWARD, 2010)

Objetivando não ser repetitivo o Scrum será melhor detalhado quando o framework proposto for descrito, sempre ressaltando as principais diferenças comparadas ao Scrum.

O Scrum é um excelente framework para projetos de desenvolvimento de software, porém para ser utilizado em outros tipos de projetos, como implementação de software, é necessário realizar algumas alterações visando melhor aderência as necessidades de mercado, terminologia e processos específicos.

Outro ponto relevante é que no Scrum os artefatos poderão ser adicionados ao framework, porém isto não é incentivado e fica a critério de cada empresa. Observando o mercado e ambiente a qual este trabalho está se destinando observar-se necessidades específicas e comuns nos projetos. Desta maneira com o objetivo de criar algo mais conciso e padronizado o framework proposto irá estabelecer um conjunto padrão de artefatos adicionais obrigatórios e sugerir alguns específicos.

3. FRAMEWORK DE GERENCIAMENTO DE PROJETOS COMPLEXOS

Assim como o SCRUM o framework proposto baseia-se em ciclos de trabalho, que é constituído por um objetivo e itens para serem entregues. Ao final de cada ciclo o trabalho é apresentado e é feito o planejamento do próximo ciclo. Este ciclo é chamado de Sprint.

O Projeto inicia-se com um escopo básico ou uma visão, que refletirá em outro componente que faz parte da espinha dorsal do SCRUM, que é o Product Backlog que aqui chamaremos de Project Backlog. É a lista priorizada de itens que deverão ser entregues no projeto. Esta lista é dinâmica e pode ser atualizada a qualquer momento pelo cliente, por este motivo, o framework é considerado preparado para mudanças de escopo, alterações de direcionamentos estratégicos, mitiga riscos através de reavaliação constantes e múltiplas interações de Sprint e principalmente foca em entregar o que tem mais valor ao negócio do cliente, antecipando os benefícios esperados do projeto.

O framework proposto também é composto de Papéis, Cerimonias e Artefatos que foram modificados e aperfeiçoados afim de atender as necessidades de um projeto de implementação de software complexo, adicionando a ele mais governança sobre o projeto bem como ajustando-o para um modelo mais efetivo de prestação de serviço.

A seguir a visão geral do framework com os seus papéis, cerimônias e artefatos.

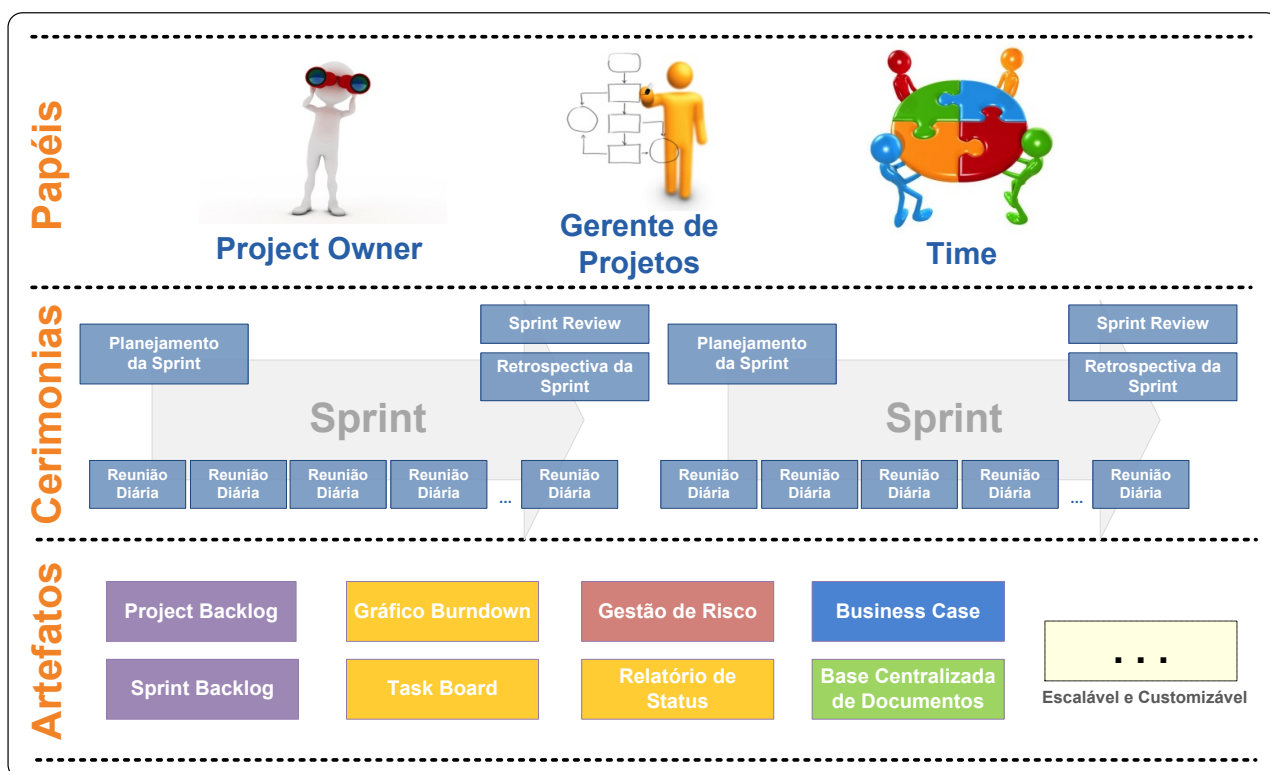


Figura 5: Visão geral do Framework

Papéis são desempenhados por pessoas que assumem uma responsabilidade dentro da organização temporária denominada projeto. O ideal que a relação seja de 1 para 1, mas uma pessoa poderá ser Gerente de Projetos em um projeto e membro do Time no outro. As Cerimônias são eventos que acontecem durante o ciclo de vida de cada Sprint, cada uma delas tem seus objetivos e resultados esperados. Os artefatos são conjuntos de ferramentas e técnicas, alguns obrigatórios outros opcionais, para suportar o ciclo de vida do projeto, aprimorar a governança e fornecer as informações necessárias para todos os papéis envolvidos e para organização. Novos artefatos poderão ser adicionados por necessidades da organização, melhoramento contínuo ou para cumprir conformidades obrigatórias.

3.1. Papéis

Project Owner: Equivalente ao Product Owner do SCRUM é o responsável pelo valor ao negócio do projeto. Obrigatoriamente um Project Owner deverá ser um responsável pelo projeto do lado do cliente, que tem a visão exata das necessidades, anseios e expectativas corporativas da empresa. Suas principais atribuições são:

- Representante oficial do cliente;
- Realizar o alinhamento do projeto com as necessidades corporativas;
- Justificar o projeto para a organização e realizar a reavaliação constante do seu valor;
- Manter o Project Backlog atualizado e priorizando as necessidades;
- Estabelecer os critérios de aceitação dos itens do Project Backlog;
- Estar disponível para o Time e o Gerente de Projetos para responder questionamentos;
- Autoridade para aceitar ou recusar os itens entregues;
- Detém autoridade para parar ou cancelar o projeto;

Gerente de Projetos: Seria o mais próximo ao papel do ScrumMaster, porém no framework SCRUM o ScrumMaster é um especialista no framework agindo mais como um facilitador, técnico e solucionador de impedimentos. No framework proposto além das atribuições equivalentes ao SCRUM este papel agirá de forma mais ativa no projeto tendo responsabilidades como:

- Certificar junto com o time que o(s) item(s) priorizado(s) no Project Backlog tem detalhamento o suficiente;
- Remover obstáculos do projeto;
- Ser facilitador na resolução de conflitos entre membros do time ou entre o Time e o Project Owner;
- Realizar e revisar a Gestão de Riscos do projeto a cada ciclo de *Sprint*;

- Comunicar-se com o Project Owner e demais partes interessadas do cliente;
- Medir a evolução do projeto;
- Fazer e distribuir o relatório de status do projeto;
- Subsidiar o Project Owner com informações relativas ao projeto quando se fizer necessário;
- Manter a base única de documentação e informações do projeto, detalhada a seguir como um artefato do projeto;
- Participar de todas as cerimônias de início e fechamento de ciclos de *Sprint* e sempre quando possível das reuniões diárias do projeto;
- Assegurar a alocação de recursos necessária para que o projeto seja executado;
- Garantir a produtividade do time;

Time: Com poucas diferenças do framework SCRUM, o time deve ser auto-organizável, com conhecimentos multidisciplinares para realização dos trabalhos necessários do projeto. O time tem autoridade para realizar o auto gerenciamento na distribuição das atividades entre seus membros e discussão de alternativas técnicas sempre focando na entrega dos itens acordados e na meta da Sprint. Junto com o Gerente do Projeto são responsáveis por criar as estimativas, reportar status, discutir impedimentos, identificar riscos e documentar o projeto.

3.2. Cerimônias

Planejamento da Sprint: Assim como a Sprint Planning do SCRUM esta reunião tem como objetivo determinar a meta da próxima Sprint e definir e planejar quais itens do Project Backlog serão executados dentro da próxima Sprint e revisar os critérios de aceitação destes itens.

Ao contrário do framework SCRUM a Sprint no framework proposto não tem tamanho fixo para um projeto, exemplo a cada 2 semanas, o tamanho da próxima Sprint é também definido nesta reunião podendo ser no mínimo de 1 semana e o seu valor máximo poderá chegar a 2 meses dependendo do grau de conforto do

Time, Gerente de Projetos, e Project Owner relacionado ao gerenciamento do projeto observando-se a complexidade das atividades a serem executadas.

Um artefato também adicionado ao framework para esta reunião é a identificação e a revisão dos riscos do projeto. O Gerente de Projetos é o responsável por esta atividade com a participação ativa de todos os membros do time e do Project Owner. Ao ser identificado um risco o mesmo deverá ser quantificado e qualificado e depois dependendo do impacto e da zona de conforto de todos, se necessário, criado um plano de resposta a este risco com as devidas contingências e planos de ação.

Reunião Diária: Assim como a cerimonia Daily Scrum é uma reunião informal, sem ATA de reunião, do time para discutir impedimentos e evolução do projeto. Ela deve acontecer todos os dias sempre no mesmo horário para facilitar e deve durar no máximo 20 minutos. Durante esta reunião cada membro do time irá dizer brevemente o que fez, o que está trabalhando, quais são suas dificuldades e impedimentos. Caso o Gerente do Projeto esteja presente, ela será o facilitador desta reunião coletando as informações de evolução do projeto, bem como obtendo os impedimentos para serem trabalhados, caso não esteja outro membro do grupo irá coletar estas informações e passar ao Gerente de Projeto. Como um time auto-organizável nesta reunião o time poderá reorganizar as atividades e atribuições dos seus membros, a fim de garantir a entrega todos os itens do Sprint Backlog e cumprir a meta da Sprint.

Sprint Review: Assim como no framework SCRUM nesta reunião todos papeis são convidados inclusive outras partes interessadas do cliente, além do Project Owner. É durante esta reunião que o time faz apresentação dos trabalhos executados na última Sprint e obtém aprovação do Product Owner. Adicionalmente ao framework SCRUM, como estamos trabalhando um modelo para área de consultoria e serviços, é no final desta cerimonia que o Gerente de Projetos obtém a assinatura formal do Project Owner para faturamento do trabalho, que poderá ser por entregáveis ou por horas trabalhadas dependendo da negociação com o cliente.

Itens que por algum motivo não sejam aceitos como entregues pelo Product Owner voltarão para a Project Backlog para serem reavaliados e priorizados. Não entrando automaticamente na próxima Sprint. Esta decisão fica a critério do Project Owner a ser avaliada em conjunto com o time e o Gerente de Projetos durante a reunião de Planejamento da Sprint.

Retrospectiva da Sprint: Assim como a cerimonia Sprint Retrospective do SCRUM, durante esta reunião o time e o Gerente de Projeto faz uma retrospectiva do que aconteceu durante a última Sprint, o que deu certo o que deu errado, pontos a serem melhorados, como devem ser melhorados sobre a ótica do que está sendo entregue do projeto e do processo de trabalho. Desta forma fechando o ciclo constante de melhoramento continuo de inspeção e adaptação.

Estas cerimoniaes são cíclicas e fazem parte de todo o ciclo de vida do projeto. As reuniões de planejamento da Sprint e Retrospectiva juntas não devem durar mais de 1 dia e meio, sendo o ideal 1 dia. A ilustração a seguir demonstra estas cerimoniaes distribuídas na linha do tempo do projeto.

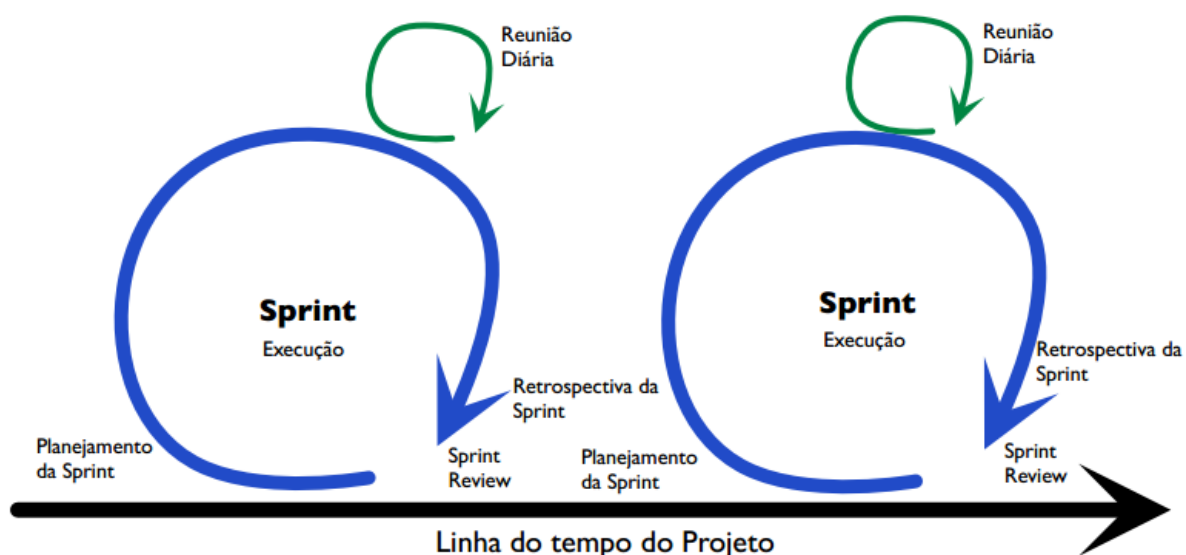


Figura 6: Cerimonias do Framework

3.3. Artefatos

Project Backlog: É a lista priorizada dos itens a serem implementados durante o projeto. A responsabilidade de mantê-la, realizar o alinhamento da mesma com os objetivos do negócio bem como prioriza-las seguindo critérios de entrega de valor é do Project Owner. Quanto mais alta for a prioridade de um item, mais chance ele tem de entrar na próxima Sprint e mais ele deve estar com seu escopo detalhado. Um item muito grande poderá ser dividido em vários itens menores para serem possíveis de serem executados em uma Sprint, observando-se sempre os critérios de conforto de gestão relacionados ao tamanho de cada Sprint, gestão e riscos.

É importante ressaltar que o projeto deverá ser finalizado quando o Product Owner ficar satisfeito com o que foi entregue e o benefício esperado for alcançado, independente se tem ou não itens pendente no Project Backlog, inclusive é comum o projeto ser finalizado com itens ainda no Project Backlog que não tem as devidas prioridades e não justificam o valor para o negócio naquele momento.

Na prática o Project Backlog poderá ser um software específico ou uma planilha eletrônica com links para documentos quando detalhes mais específicos ou documentações se fizerem necessário.

Sprint Backlog: É o conjunto de trabalho proveniente do Project Backlog acordado pelo time para ser entregue durante uma Sprint detalhados por atividades. Este Sprint Backlog é acordado e desenvolvido durante a cerimônia de Planejamento da Sprint e na metodologia proposta ele deve conter no mínimo, os itens que foram selecionados para a Sprint com suas respectivas atividades macros para serem executadas pelo time, dono da atividade e tempo estimado. O time é auto-organizável e tem autoridade para durante uma Sprint alterar, adicionar ou excluir atividades, bem como, alterar seus donos visando sempre cumprir o objetivo da Sprint de entregar os itens selecionados do Project Backlog. Estes ajustes são normalmente feitos durante a reunião diária.

Se por algum motivo de problema técnico, risco, impedimento e até mesmo por causa de estimativa incorreta de tempo de execução o time observa que não conseguirá entregar na íntegra um item, o Gerente de Projetos deve comunicar o Project Owner o mais breve possível, porém o item não será removido do Sprint Backlog e voltará para o Project Backlog no final da Sprint.

Gráfico Burndown: Gráfico de acompanhamento do andamento da Sprint baseado no tempo de trabalho restante. Este gráfico é atualizado de forma rápida e objetiva com a participação de todos os membros do time durante a Reunião Diária, trabalho este feito pelo Gerente de Projetos ou por um membro designado. Trata-se da principal ferramenta do time do projeto para acompanhar o andamento da Sprint, uma vez que as atividades não necessitam serem executadas em ordem e faz parte da cultura de auto-organização a mudança e adaptação constante de atividades e responsabilidades, sempre mantendo o foco na meta da Sprint.

O Gráfico Burndown inicia-se ao final do planejamento de uma Sprint, onde obtêm-se as atividades dentro do Sprint Backlog e é somado o tempo total de execução previsto, formando o eixo Y. O eixo X é a quantidade de dias da Sprint.

Com foco em produtividade e no cumprimento dos objetivos da Sprint, não é medido a quantidade de trabalho expendido para execução de uma atividade e sim tempo estimado para sua conclusão, independentemente da previsão inicial durante o Planejamento da Sprint. Somando este tempo restante estimado, atualizado diariamente, temos o gráfico Burndown.

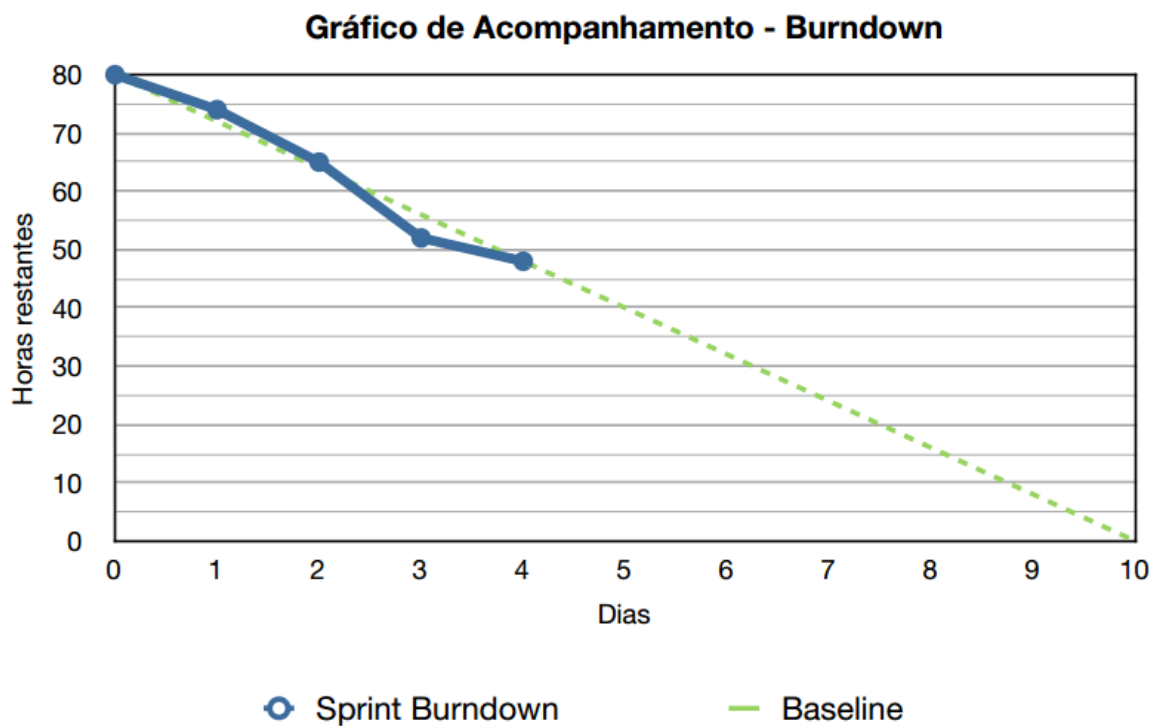


Figura 7: Exemplo de Gráfico de Acompanhamento – Burndown

De forma simples e sucinta, a linha azul estando acima da linha verde significa que a Sprint está atrasada e abaixo adiantada. O ideal é que o gráfico fique sempre o mais perto da linha base possível. Derivações grandes neste gráfico podem significar problemas ou estimativas incorretas e deve ser pauta para discussão durante a reunião de Retrospectiva da Sprint, visando sempre otimização de recursos e melhoramento contínuo.

Task Board: O Task Board não é um artefato do SCRUM, mas muitas equipes que trabalham com este framework usam. Devido ao incontestável poder de comunicação que este artefato propicia ele foi adicionado a esta proposta de framework.

Basicamente o Task Board é um painel colocado na parede (*flipchart*, cartolina, ou quadro branco) disponível na sala de trabalho da equipe contendo as atividades (escritas ou em Post-It®) a serem executadas dentro de uma Sprint para que os objetivos da mesma e todos os itens selecionados sejam entregues. Estas

atividades são classificadas em uma das três categorias, Para fazer (pendentes), Fazendo (em execução) ou Feita (pronta).

Assim como o artefato Gráfico Burndown, o Task Board serve para manter todos atualizados e informados sobre o andamento dos trabalhos de uma Sprint e deverá ser atualizado durante a Reunião Diária. O elemento visual é importante e este quadro deve ficar visível e acessível a todos os membros do time, motivando a todos a levar suas atividades para o lado direito do quadro e cumprir os objetivos da Sprint.

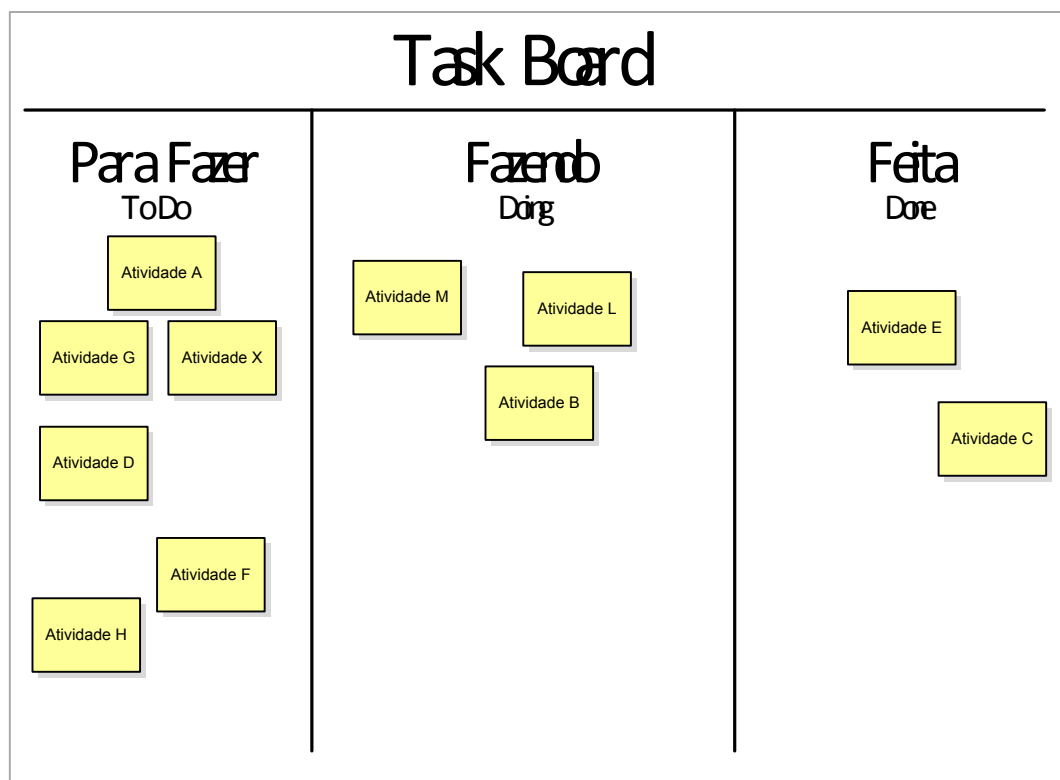


Figura 8: Exemplo de Task Board (SIMS; JOHNSON, 2011)

Gestão de Riscos: Este artefato foi adicionado ao framework de Gestão de Projetos complexos visando aumentar a governança, transparência, previsibilidade e maximizar a entrega de valor do projeto com o mínimo de desvio possível antecipando-se o planejamento das ações e planos de respostas antes que o evento ocorra.

Segundo o PMI “risco é um evento ou uma condição incerta que, se ocorrer, tem um efeito em pelo menos um objetivo do projeto. Os objetivos podem incluir escopo, cronograma, custo e qualidade. Um risco pode ter uma ou mais causas e, se ocorrer, pode ter um ou mais impactos” (PMI, 2008, p. 275).

No framework a gestão de riscos é feita de maneira bem dinâmica e simples, mantendo o foco no que é mais importante. Concentrando-se em levantamento dos riscos, qualificação referente a probabilidade de acontecimento, quantificação relacionando o seu impacto direto ao projeto e estratégias de respostas que serão elaboradas durante a cerimônia de Planejamento da Sprint que poderão ser classificadas como aceitar, mitigar, transferir ou eliminar o risco para ameaças e explorar, compartilhar, melhorar e aceitar para oportunidades (PMI, 2008; p. 303) .

A função de guiar a gestão de riscos e instigar os membros do time e o Project Owner a contribuir é do Gerente de Projetos.

Relatório de Status: Como este framework foi baseado em metodologia ágeis que valoriza mais a entrega do projeto do que a documentação e processos sistemáticos, este item é opcional. Ele foi incluído ao framework, pois muitos clientes solicitam relatório de status executivo do projeto para ser entregue a outras partes interessadas dentro da organização. Este relatório será confeccionado pelo Gerente de Projetos com apoio direto do Project Owner que detém o conhecimento e a visão de entrega de valor para a organização.

Seu formato, *layout*, forma de distribuição e publicação será discutido e acordado entre o Gerente de Projetos e o Project Owner, mas minimamente contem um sumário do que já foi entregue, a quantidade de recursos utilizada, o que está sendo planejado para ser entregue na Sprint atual e o seu status.

Business Case: O Business Case também foi incorporado a este framework com o objetivo de criar um norte para o projeto, mitigar a dispersão do foco e principalmente apoiar o Project Owner a justificar continuamente o valor do projeto para negócio do cliente.

Segundo a publicação mais recente da metodologia de gestão de projetos PRINCE2™, que tem como princípio número um justificativa continua para o negócio, o objetivo do Business Case é estabelecer mecanismos para avaliar se o projeto é (e continua sendo) desejável, viável, realizável e também como um meio para apoiar a tomada de decisão (continuada) em seus investimentos (OGC, 2009, p. 21).

O Business Case pode e deve ser alterado durante o projeto se as variáveis internas e externas do ambiente do cliente mudarem fazendo com que o projeto mude o seu direcionamento e precise ser reavaliado. Ele deve servir como principal suporte em qualquer tomada de decisões do projeto e é a mais importante ferramenta do Project Owner para definir se o projeto pode ser finalizado pois atingiu os objetivos estratégicos ou deve ser cancelado pois seu valor não é mais justificável ao projeto e não faz sentido continua-lo.

Desenvolver o Business Case é uma função do Project Owner do projeto com apoio, suporte e direcionamento do Gerente de Projetos. Este artefato apesar de ser fortemente recomendado não é obrigatório no framework uma vez que algumas empresas optam por não investir seu tempo e dinheiro para fazer um Business Case para um projeto de implementação de software. Quando isto acontecer o Project Owner e o Gerente de Projetos devem ter atenção redobrada para evitar a dispersão do foco do projeto.

Base Centralizada de Documentos: Este artefato foi adicionado ao framework para facilitar a comunicação entre todos os membros do projeto e é um item obrigatório, incrementando a gestão, controle e distribuição de informações. Todos os documentos citados no framework, tais como Project Backlog, Sprint Backlog, Riscos, Gráfico Burndown, Relatório de Status, entre outros; deverão estar armazenados e disponíveis a todos os membros do projeto nesta base centralizada, que preferencialmente deverá possuir tecnologia de controle de versões.

4. ALINHAMENTO COMERCIAL

Um framework de gerenciamento de projetos complexos para implementação de software como proposto neste trabalho, para ser aplicado, é crucial ter um alinhamento com a área comercial.

Analisando os modelos atuais de negociações de contrato de serviços relacionados a implementação de software, existe dois modelos de contratação básico:

Horas – O cliente pagará o projeto por hora de trabalho da equipe de consultoria, independente do que foi entregue.

Entregáveis – O cliente pagará o projeto por entregáveis definidos no contrato, independente da quantidade de horas e recursos que foram consumidos.

Ambos os modelos tem vantagens e desvantagens e riscos divergentes. Por exemplo, o modelo de contratação por Horas poderá ser superestimado ou subestimado e não atender as expectativas do cliente quanto a entrega de valor e suas necessidades, criando frustração e desavenças. Além disso, o risco maior fica do lado do cliente que pode pagar mais dinheiro do que previa para concluir o projeto. Já o modelo de Entregáveis está fortemente relacionado a uma boa análise do ambiente do cliente e um bom levantamento de requisito e escopo do projeto, caso contrário o fornecedor está sujeito ao risco direto de gastar muito mais do que previa para executar o trabalho, ocorrer desavenças e discussões com o cliente sobre o que faz parte e o que não faz parte do escopo do projeto, e não entregar o valor esperado porque o mesmo não foi corretamente definido e identificado no início das negociações, deixando o contrato pouco flexível.

Outro ponto pouco observado quando estamos negociando um contrato de serviços está relacionado a satisfação do cliente.

Muitas empresas estão preocupadas em vender seu serviço de consultoria o máximo que puder para que o cliente pague mais e ela tenha um bom faturamento. Outras que tem o foco apenas na venda de software, mas tem sua área de consultoria apenas por necessidade para manter seu negócio, querem que o serviço seja executado o quanto antes e que use o menor número de recursos possíveis. Poucas observam o fator satisfação do cliente, que está diretamente ligado a necessidade do negócio ser atendida. Como exemplo, muitos clientes contratam uma solução inteira de gestão de negócio quando na realidade gostaria apenas de controlar o estoque e gerar nota fiscal eletrônica.

Em muitos casos itens não relevantes para o cliente geram atrasos no projeto, prolongando o mesmo, gastando mais tempo e dinheiro desnecessariamente.

Jesse Fewell alinhado com metodologias ágeis defende mais resultados para o negócio com menos trabalho possível e complementa, “A causa mais dramática de atrasos nos projetos é que estamos fazendo mais trabalho do que é absolutamente necessário apenas para atingir a declaração de escopo que foi definida.” (FEWELL, 2012)

O framework proposto também ajuda neste processo uma vez que as entregas são incrementais e mudanças são perfeitamente aceitas, pois o Project Backlog está em um processo constante de evolução, alteração e priorização sobre o controle direto do cliente, sempre avaliado sobre a ótica da necessidade e entrega de valor ao negócio. O contrato poderá ser negociado inicialmente por Sprint e uma vez que o cliente ganhe confiança na metodologia utilizada poderá ser alterado por hora, uma vez que ele sempre vai ter controle do que está sendo executado e planejado. Garantindo assim entrega antecipada e economia de recursos.

5. CONCLUSÃO

Diante das pressões corporativas para fechamento precipitado do contrato, uso de orçamento remanescente, falta de agenda dos consultores e cliente, omissões, lacunas de informações e falta de compreensão dos objetivos estratégicos, resultam em um grande desafio para empresas implementarem uma solução de software.

Metodologias de Gestão de Projetos tradicionais, tais como PRINCE2 e as criadas baseada no PMI, são excelentes e recomendadas, porém intensamente dependente de informações e definições claras. Estas metodologias tem por característica uma importante fase inicial de planejamento que tem como produto o plano do projeto. Justamente a ausência destas informações e definições logo no início do projeto torna-se complexo a confecção de um bom plano de projeto e por consequência a gestão do projeto com modelos tradicionais.

Assim, o escopo mal definido, ausência de levantamento de riscos, definição de cronograma carente, inexistência de padrões e critérios de qualidade e principalmente não identificação precisa das expectativas, justificativas, objetivos e valores para o negócio do cliente, culmina com um plano de projeto excessivamente macro, irreal e que não atende as necessidades e anseios do cliente. Desta forma, o sucesso do projeto e satisfação do cliente torna-se exceção e as possibilidades de frustração e descontentamento de ambas as partes é eminente.

O modelo proposto é uma adaptação do conhecido framework de desenvolvimento de software SCRUM, adaptado e estendido para suportar projetos de implementação de software e foi especialmente aperfeiçoado para trabalhar com estes cenários. Através de interações múltiplas, maior participação do cliente, trabalhos incrementais, antecipação de benefícios, mitigação de riscos por interações múltiplas, receptividade a grandes mudanças, flexibilidade e prioridade a entrega de valor ao cliente, faz do modelo proposto capaz de lidar com os desafios supracitados.

O framework é muito simples, mas não significa que seja simples implementá-lo. A equipe deverá ser treinada e acreditar na eficiência do modelo interativo, com processo empírico e flexível. O cliente deverá entender o framework, aceitar condições contratuais de pagamento baseado no modelo proposto e nomear um *Project Owner* que conheça realmente as necessidades da organização, tenha bom relacionamento interno, esteja disposto a assumir a responsabilidade de guiar o projeto em linha com a visão corporativa.

O sucesso do projeto não é determinado pela metodologia ou framework utilizado e sim pelo seu resultado e valor entregue a organização. Um bom gestor de projetos sabe identificar as dificuldades e necessidades de cada cliente e definir a metodologia de trabalho que irá melhor adaptar-se em cada situação. Nesse sentido, este trabalho apresentou de forma objetiva e padronizada um modelo testado pelo autor, com sucesso e com alto índice de satisfação, mais uma alternativa aos gerentes de projetos, especialmente quando se depararem com situações onde o cenário é muito dinâmico, o escopo mal definido e os riscos elevados.

REFERÊNCIAS

FEWELL, Jesse; matéria **Less is more**; Revista PM Network, Fevereiro 2012 volume 26 número 2, PMI, tradução nossa.

HASS, Kathleen B. **Managing Complex Projects: A New Model**, 2009. Capítulo 3, tradução nossa.

HIGHSMITH, Jim. **History: The Agile Manifest**, 2001, <http://agilemanifesto.org/history.html>, acessado em agosto de 2012.

KERZNER, Harold. **Project Management Metrics, KPIs and Dashboards**, 2011. Capítulo 1.

OGC®. **Managing Successful Projects with PRINCE2™** Quinta Edição, 2009. p. 1, tradução nossa.

OGC®. **Managing Successful Projects with PRINCE2™** Quinta Edição, 2009. p. 21, tradução nossa.

PMI®. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)** Quarta Edição, 2008. p. 5.

PMI®. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)** Quarta Edição, 2008. p. 275.

PMI®. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)** Quarta Edição, 2008. p. 303.

PMI®. **Um Guia do Conjunto de Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)** Terceira Edição, 2004. p. 375.

PRIES, Kim; QUIGLEY, Jon. **Scrum Project Management**, 2011, Capítulo 1.4, tradução nossa.

QSM Associates, Inc. **The Agile Impact Report**, Maio de 2008.

RICHES, Tony. **Agile Project Management for Busy Managers**, 2012. Capítulo 1, tradução nossa.

SCHWABER, Ken; SUTHERLAND, Jeff. etc; **Manifesto para Desenvolvimento Ágil de Software**, 2001, <http://agilemanifesto.org/iso/ptbr/>, acessado em Agosto de 2012.

SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum / The Scrum Guide**, Outubro de 2012.

SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum / The Scrum Guide**, Outubro de 2012, p 17.

Scrum Alliance®. **The Scrum: Basics – Scrum 101**, http://www.scrumalliance.org/pages/scrum_101, acessado em Agosto de 2012.

SIMS, Chris; JOHNSON, Hillary. **The Elements of Scrum**, Primeira edição, 2011. Parte 1, tradução nossa.

SIMS, Chris; JOHNSON, Hillary. **The Elements of Scrum**, Primeira edição, 2011. Capítulo Scrum Artifacts, modelo baseado.

STACEY, Ralph D., **Strategic Management and Organisational Dynamics: The Challenge of Complexity**, Primeira Edição, 2000. Pearson Education, Inc, modelo adaptado.

STEPANEK, George. **Software Project Secrets: Why Software Projects Fail** 2005, p. 8, tradução nossa.

WOODWARD, Elizabeth; SURDEK, Steffan; GANIS, Matthew. **A Practical Guide to Distributed Scrum**, 2010, IBM Press – Primeira Edição, p 4.

GLOSSÁRIO

OGC® – É uma organização independente do governo da Inglaterra, que tem como objetivo melhorar a eficiência e eficácia de compras governamentais e promover a participação do setor privado em todo o setor público inglês.

PMBOK® – Conjunto de conhecimentos, práticas e ferramentas de gestão de projetos publicado pelo PMI (Project Management Institute – www.pmi.org), atualmente na sua quarta edição.

PMI® – Fundada em 1969, uma organização não lucrativa que tem o intuito de desenvolver e divulgar métodos de desenvolvimento de projetos. Maiores informações www.pmi.org

PRINCE2™ - Criado em 1989 a partir da metodologia PROMPTII (1975) e lançado em 1996 o PRINCE2™ é uma metodologia de gerenciamento de projetos. PROMPTII e PRINCE2™ são padrões nos projetos de sistemas de informação do Governo Britânico desde 1979. Sua referencia oficial atual é o livro “Managing Successful Projects with PRINCE2™” publicado em 2009 pela TSO – ISBN 978 0 11 331059 3.

SCRUM – O termo varia do nome de uma jogada de Rugby onde duas equipes disputam posse da bola. O Scrum é um framework para desenvolver e manter produtos complexos.

Sprint – Um ciclo de trabalho que tem um prazo definido, no Scrum ele tem um tempo fixo para um projeto, exemplo ciclos de 2 em 2 semanas.

No framework proposto este tempo do ciclo é negociado durante a reunião de planejamento da Sprint.